UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/802,952	03/12/2001	Junya Yamashita	60188-042	4805

7590 06/22/2005

Jack Q. Lever, Jr.
McDERMOTT, WILL & EMERY
600 Thirteenth Street, N.W.
Washington, DC 20005-3096

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 06/22/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/802,952	YAMASHITA ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 April 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 May 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

pd

DETAILED ACTION

1. Claims 1-21 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 4/25/2005.

Specification

3. The title of the invention is not descriptive and is not clear to the examiner (what is "branch processing of shifting control"?). A new title is required that is clearly indicative of the invention to which the claims are directed. The examiner asserts that some reference should be made to detecting a last instruction before a branch and then retrieving the first instruction after the branch for execution, as this concept appears to be in each of the claimed embodiments. Applicant is reminded that MPEP 606.01 states "This may result in slightly longer titles, but the loss in brevity of title will be more than offset by the gain in its informative value in indexing, classifying, searching, etc. If a satisfactory title is not supplied by the applicant, the examiner may, at the time of allowance, change the title by examiner's amendment."

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims **1-3, 5, 7-10, 12-15,** and **21** are rejected under 35 U.S.C. 102(b) as being anticipated by Jaggar, U.S. Patent No. 5,506,976 (as applied in the previous Office Action).

6. In regard to claim 1:

7. Jaggar discloses a processor (Fig. 2) comprising:

a) instruction executing means (fig. 2, processing pipeline 2) for executing an instruction stored in storing means (col. 6, lines 44-45, fig. 2, element 8);

b) execution instruction address outputting means (fig. 2, program counter 10 outputs address to be fetched to memory 8) for outputting an execution instruction address that is an address of an area in which an instruction to be executed by the instruction executing means is stored;

c) detecting means (fig. 2, program counter 10, reach value register 16, comparator 14, and branch cache 4) for detecting that the instruction to be executed is a last instruction of a process in processing before branching (in fig. 1, instruction E is shown to be the last instruction of a process before branching; initially instruction E is detected as the last instruction before branching by detecting the branch instruction by the branch detection means 22, whereby an entry in the branch cache 4 is created for instruction E [col. 7, lines 39-49], subsequently instruction E is detected by comparing it with the entries in the branch cache [col. 8, lines 15-25] i.e. branch cache hit), the detection being performed by comparing information on the instruction to be executed to predetermined information on the process one by one. See Fig.3, Fig.4, and column 8, lines 1-16. Note that each

instruction address is compared to the predetermined reach value. Consequently, the comparisons occur one by one (i.e., one instruction at a time).

d) wherein the execution instruction address outputting means outputs a start address (&U, fig. 3) that is an address of an area in the storing means in which a first instruction of a process after branching (U, fig. 1) is stored, when the last instruction (E, fig. 1) is detected by the detecting means (col. 8, lines 26-31).

8. In regard to claim 2:

9. Jaggar has taught a processor as described in claim 1. Jaggar has further taught that the execution instruction address outputting means comprises:

a) start address storing means (fig. 2, branch cache 4) for storing a start address (target addresses) of each of a plurality of processes in the storing means;

b) start address selecting means for sequentially switching and selecting the start address stored in the start address storing means, every time the last instruction is detected by the detecting means (when a branch cache hit occurs, a start address [target address] associated with that entry is selected col. 8, lines 31-35),

c) wherein the execution instruction address is output based on the start address selected by the start address selecting means (fig. 4, the program counter 10 receives the start address from the target address latch 18 col. 8, lines 26-31).

10. In regard to claim 3:

11. Jaggar has taught a processor as described in claim 1. Jaggar has further taught:

a) end address storing means (fig. 2, branch cache 4) for storing an end address (reach value, col. 6, lines 50-51) that is an address of an area in which a last instruction of each of a plurality of processes is stored in the storing means (in the example given in fig. 3, the address of the last instruction E of a process [see fig. 1] is stored in the reach value of Q' entry);

b) end address selecting means for sequentially switching and selecting the end address stored in the end address storing means, every time the last instruction is detected by the detecting means (when a branch cache hit occurs, an end address [reach value] associated with that entry is selected col. 8, lines 26-28),

c) wherein the detecting means detects the last instruction (col. 6, lines 64-67) based on the execution instruction address output from the execution instruction address outputting means (fig. 2, program counter 10) and the end address selected by the end address selecting means (reach value register 16).

12. In regard to claim 5:

13. Jaggar discloses a processor (Fig. 2) comprising:

a) instruction executing means (Fig. 2, processing pipeline 2) for executing an instruction stored in storing means (col. 6, lines 44-45, Fig. 2, element 8);

b) execution instruction address outputting means (Fig. 2, program counter 10 outputs address to be fetched to memory 8) for outputting an execution instruction address that is an address of an area in which an instruction to be executed by the instruction executing means is stored;

c) the processor further comprising:

d) detecting means (Fig. 2, program counter 10, reach value register 16, comparator 14, and branch cache 4) for detecting that the instruction to be executed by the instruction executing means is a last instruction of a process before branching (in Fig.1, instruction E is shown to be the last instruction of a process before branching; initially instruction E is detected as the last instruction before branching by detecting the branch instruction by the branch detection means 22, whereby an entry in the branch cache 4 is created for instruction E [col. 7, lines 39-49], subsequently instruction E is detected by comparing it with the entries in the branch cache [col. 8, lines 15-25] i.e. branch cache hit);

e) wherein the execution instruction address outputting means outputs a start address (&U, Fig.3) that is an address of an area in the storing means in which a first instruction of a process after branching (U, Fig.1) is stored, when the last instruction (E, Fig.1) is detected by the detecting means (col. 8, lines 26-31).

f) wherein the detecting means detects the last instruction based on judgment whether information stored in correspondence with information indicating a content of an instruction to be executed by the instruction executing means in the storing means indicates the last instruction. See Fig.3, Fig.4, and column 8, lines 1-16. Note that each instruction address is compared to the predetermined reach value and also to a tag in the cache. The reach value and the tag are stored data which indicate the last instruction. Consequently, if the comparator determines a match has occurred and a cache hit occurs, then the last instruction before a branch is detected.

14. In regard to claim 7:

15. Jaggar has taught a processor as described in claim 2. Jaggar has further taught that the start address storing means comprises a memory storing the start address (branch cache 4).

16. Although Jaggar does not explicitly mention that the start address selecting means comprises address designating means for designating an address of an area in which the start address is stored in the memory, it is deemed inherent to the branch cache memory to have an address designating means to address a location in the cache. Otherwise, the data could not be retrieved from it.

17. In regard to claim 8:

18. Jaggar has taught a processor as described in claim 2. Jaggar has further taught that the start address stored in the start address storing means can be set by execution of an instruction by the instruction executing means (target address is stored on execution of a branch instruction col. 7, lines 47-52).

19. In regard to claim 9:

20. Jaggar has taught a processor as described in claim 2. Jaggar has further taught that the start address stored in the start address storing means can be set by a supervisory processor (fig .2, branch instruction detector 22 sets the target address col. 7, lines 39-52) for controlling an operation of the processor.

21. In regard to claim 10:

22. Jaggar has taught a processor as described in claim 9. Jaggar has further taught start address storing means (fig. 2, branch cache 4) for the supervisory processor (fig. 2, branch instruction detector 22) for storing a start address output from the supervisory processor,

wherein the start address stored in the start address storing means for the supervisory processor is written in the start address storing means at predetermined timing (when the branch instruction's target address is determined, col. 7, lines 48-52).

23. In regard to claim 12:

24. Jaggar has taught a processor as described in claim 3. Jaggar has further taught that the end address storing means comprises a memory storing the end address (branch cache 4).

25. Although Jaggar does not explicitly mention that the end address selecting means comprises address designating means for designating an address of an area in which the end address is stored in the memory, it is deemed inherent to the branch cache memory to have an address designating means to address a location in the cache. Otherwise, the data could not be retrieved from it.

26. In regard to claim 13:

27. Jaggar has taught a processor as described in claim 3. Jaggar has further taught that the end address stored in the end address storing means can be set by execution of an instruction by the instruction executing means (reach value, PC-1, is stored on execution of a branch instruction col. 7, lines 47-49).

28. In regard to claim 14:

29. Jaggar has taught a processor as described in claim 3. Jaggar has further taught that the end address stored in the end address storing means can be set by a supervisory processor (fig .2, branch instruction detector 22 sets the reach value, PC-1, col. 7, lines 39-48) for controlling an operation of the processor.

30. In regard to claim 15:

31. Jaggar has taught a processor as described in claim 14. Jaggar has further taught end address storing means (fig. 2, branch cache 4) for the supervisory processor (fig. 2, branch instruction detector 22) for storing an end address output from the supervisory processor,

wherein the end address stored in the end address storing means for the supervisory processor is written in the end address storing means at predetermined timing (when the branch instruction is detected, col. 7, lines 39-48).

32. In regard to claim 21:

33. Jaggar has taught a processor as described in claim 5. Jaggar has further taught that the detecting means detects by comparing information corresponding to the instruction to be executed by the execution means and a predetermined information in accordance with the process in execution one by one. See Fig.3, Fig.4, and column 8, lines 1-16. Note that each instruction address is compared to the predetermined reach value. Consequently, the comparisons occur one by one (i.e., one instruction at a time).

Claim Rejections - 35 USC § 103

34. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

35. Claims **4** and **17-20** are rejected under 35 U.S.C. 103(a) as being unpatentable over Jaggar, as applied above, in view of Nair, U.S. Patent No. 6,304,962 (as applied in the previous Office Action, and further in view of Atkins et al., U.S. Patent No. 5,898,866 (as applied in the previous Office Action and herein referred to as Atkins).

36. In regard to claim 4:

37. Jaggar has taught a processor as described in claim 1. Jaggar differs from the current invention in that it does not have a processing length storing means and a processing length selecting means. Furthermore, while the processor of Jaggar does have an instruction address generating means (fig. 2, program counter 10 outputs address to be fetched to memory 8) and a last instruction detecting means (fig. 2, program counter 10, reach value register 16, comparator 14, and branch cache 4), it does not disclose that the instruction address is generated by adding the start address and a relative address and that the last instruction is detected based on the relative address and the processing length selected.

38. However, Nair teaches of storing the processing length (run length) of a process (superblock) in a superblock target buffer (col. 5, lines 39-46) on the execution of a taken branch instruction (col. 6, lines 64-67; col. 7, lines 1-2). This length is a relative address of an end

address to a start address for each plurality of processes (col. 7, lines 1-3). Fetching is continued until the end address, which is calculated by adding the starting address and the process length (col. 7, lines 6-10).

39. Atkins et al. teach of a relative addressing scheme in which the execution instruction address outputting means adds a starting address (base and index register contents col. 7, lines 63-64) of a process under processing and a relative address (displacement field col. 7, line 64) of the execution instruction address to generate the execution instruction address (effective address col. 7, line 65).

40. Hence, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the processor of Jaggar by replacing the end address (return value in the branch cache 4) storing means with a process length storing means and replacing the end address selection means with a processing length selecting means for sequentially switching and selecting the processing length stored in the processing length storing means, every time the last instruction is detected by the detection means. Furthermore, it would have been obvious to one of ordinary skill in the art at the time of the invention to use relative addressing to generate an execution instruction address by adding the starting address and a relative address and a detecting means which detects the last instruction based on the relative address and the processing length (to calculate the end address) selected by a processing length selecting means.

41. One of ordinary skill in the art at the time of the invention would have been motivated to use the length of a process in place of the end address because it occupies less space and hence translates to savings in hardware. Furthermore one of ordinary skill in the art at the time of the invention would have been motivated to use relative addressing to generate the execution

instruction address because a relative address is smaller than an effective address thus translating in smaller instruction length.

42. In regard to claim 17:

43. Jaggar in view of Nair and further in view of Atkins has taught a processor as described in claim 4. Jaggar in view of Nair and Atkins discloses that the processing length storing means comprises a memory storing the processing length (branch cache 4 of Jaggar).

44. Although Jaggar in view of Nair and Atkins does not explicitly mention that the processing length selecting means comprises address designating means for designating an address of an area in which the processing length is stored in the memory, It is deemed inherent to the branch cache memory to have an address designating means to address a location in the cache. Otherwise, the data could not be retrieved from it.

45. In regard to claim 18:

46. Jaggar in view of Nair and further in view of Atkins has taught the processor of claim 4. Nair further teaches that the processing length stored in the processing length storing means can be set by execution of an instruction by the instruction executing means (run length is stored on execution of a taken branch instruction col. 6, lines 64-67; col. 7, lines 1-2).

47. In regard to claim 19:

48. Jaggar in view of Nair and further in view of Atkins has taught the processor of claim 4. Nair further teaches that the processing length stored in the processing length storing means can

be set on detection of the last instruction (col. 6, lines 64-67; col. 7, lines 1-2). Although Nair does not explicitly mention that the processing length is set by a supervisory processor, it is deemed inherent to the design that a control logic exists to set the processing length which can include a supervisory processor.

49. In regard to claim 20:

50. Jaggar in view of Nair and further in view of Atkins has taught the processor of claim 19.

51. Jaggar in view of Nair and Atkins further teaches processing length storing means (branch cache 4 of Jaggar) for the supervisory processor for storing a processing length output from the supervisory processor,

wherein the processing length stored in the processing length storing means for the supervisory processor is written in the processing length storing means at predetermined timing (when the branch instruction is detected, col. 6, lines 64-67; col. 7, lines 1-2 of Nair).

52. Claim **16** is rejected under 35 U.S.C. 103(a) as being unpatentable over Jaggar in view of Nair in view of Atkins, as applied above, and further in view of Breeding, "Microprocessor System Design Fundamentals," Prentice Hall, pp.6-7, 1995, (as applied in the previous Office Action and herein referred to as Breeding).

53. **In regard to claim 16:**

54. Jaggar in view of Nair and further in view of Atkins has taught a processor as described in claim 4. Although Jaggar in view of Nair and Atkins does not teach that the selecting means

comprises of a selector explicitly, a selector would be inherent to perform the selecting means function of selecting a processing length.

55. Jaggar in view of Nair and Atkins differs from the present invention because the processing length storing means comprises of a cache memory (branch cache 4) and not a plurality of registers each of which stores the processing length.

56. Breeding teaches that registers are used for high-speed storage (pg. 7, line 8 under sec. 2.2).

57. Hence, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a plurality of registers to store the processing lengths in place of a cache memory.

58. One of ordinary skill in the art at the time would have been motivated to use registers for storing the processing lengths because they are a form of high-speed storage, which in turn would result in better performance.

59. Claims **6** and **11** is rejected under 35 U.S.C. 103(a) as being unpatentable over Jaggar, as applied above, in view of Breeding, as applied above.

60. In regard to claim 6:

61. Jaggar has taught a processor as described in claim 2. Jaggar has further taught that a start address selecting means for sequentially switching and selecting the start address stored in the start address storing means (when a branch cache hit occurs, a start address [target address] associated with that entry is selected col. 8, lines 31-35)

62. Although Jaggar does not teach that the selecting means comprises of a selector explicitly, a selector would be inherent to perform the selecting means function of selecting a start address.

63. Jaggar differs from the present invention because the start address storing means comprises of a cache memory (branch cache 4) and not a plurality of registers each of which stores the start address.

64. Breeding teaches that registers are used for high-speed storage (pg. 7, line 8 under sec. 2.2).

65. Hence, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a plurality of registers to store the start addresses in place of a cache memory.

66. One of ordinary skill in the art at the time would have been motivated to use registers for storing the start addresses because they are a form of high-speed storage, which in turn would result in better performance.

67. In regard to claim 11:

68. Jaggar has taught a processor as described in claim 3. Jaggar has further taught an end address selecting means for sequentially switching and selecting the end address stored in the end address storing means (when a branch cache hit occurs, an end address [reach value] associated with that entry is selected col. 8, lines 26-28)

69. Although Jaggar does not teach that the selecting means comprises of a selector explicitly, a selector would be inherent to perform the selecting means function of selecting an end address.

70. Jaggar differs from the present invention because the end address storing means comprises of a cache memory (branch cache 4) and not a plurality of registers each of which stores the end address.

71. Breeding teaches that registers are used for high-speed storage (pg. 7, line 8 under sec. 2.2).

72. Hence, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a plurality of registers to store the end addresses in place of a cache memory. One of ordinary skill in the art at the time would have been motivated to use registers for storing the end addresses because they are a form of high-speed storage, which in turn would result in better performance.

Response to Arguments

73. Applicant's arguments filed on April 25, 2005, have been fully considered but they are not persuasive.

74. Applicant argues the novelty/rejection of claim 1 on pages 9-10 of the remarks, in substance that:

> "As a preliminary matter, it is not entirely understood which limitations the Examiner alleges as not appearing in the claims. In fact, it appears the Examiner has misunderstood Applicants' previous arguments. Specifically, in the previous response, Applicants were NOT arguing that the present invention requires a preliminary comparison. Instead, Applicants were arguing that the present invention does not require a preliminary comparison as demonstrated by Jaggar, in which the comparator 14 spec6cally compares the reach value R currently stored in the reach value latch 16 with the lower order bits (i.e., lowermost eight bits) of the program counter value PC in the 16-bit addresses stored in the program counter register 10. In this regard, the comparator 14, at best, merely compares the contents of the reach value latch 16 with the program counter value PC so as to determine whether to hold the enable signal in the OFF state. That is, the alleged detection or comparison performed by the comparator 14 of Jaggar is merely a preliminary comparison and is actually irrelevant to the actual determination of the last instruction. In other words, Jaggar is completely silent as to utilizing a comparator that can determine whether the comparison performed therein is the last instruction."

75. These arguments are not found persuasive for the following reasons:

a) When the examiner stated that applicant is arguing limitations which are not in the claims, what was meant is that there is no language within the claim which excludes some preliminary comparison. And, even if such language were included, Jaggar still anticipates applicant's claim because the applicant's comparison is equivalent to Jaggar's reach-value and tag comparisons. That is, one by one, a portion of the PC is compared to the reach value 16 (Fig.4). If a match exists, then the PC is compared to the branch tags in the cache. If a hit occurs, then a last instruction before a branch has been detected. It should be noted that these two steps (reach value comparison and tag value comparison) are combined and referred to as the "comparison". It should further be noted that this comparison happens for every instruction (one by one) with process information (reach value).

76. Applicant argues the novelty/rejection of claim 1 on pages 10-12 of the remarks, in substance that:

> "...the Examiner asserted, "it can be seen that each and every address is compared to a reach value" and "so address &E corresponds to the last instruction before a branch if a cache hit occurs."
>
> However, it is respectfully submitted that this statement as a prelude to the rejection is not a proper basis for rejecting Applicants' claims, as the statement is directed to the Examiner's opinion rather than what is taught by the prior art. It is submitted that the "Examiner's opinion" cannot be relied on to replace the deficiency of a prior art reference.
>
> Even assuming arguendo that the Examiner's opinion has merit, this incomplete analysis overlooks and ignores the fact that Jaggar expressly discloses that if the lowermost eight bits of the program counter value PC match the value in the reach value latch 16, the enable signal is asserted ON and the value of the full instruction address from the program counter register 10 is compared with the cache tags (see, col. 8, lines 2 1-25). That is, if the program counter value PC is found to match with the reach value R, another comparison is performed so as to determine whether a branch cache hit has occurred. Indeed, it is respectfully submitted that only after the comparator 14 has performed the necessary comparison between the reach value R in the reach value latch 16 and the lowermost eight bits of the program counter value PC stored in the program counter register 10 will there be an ON enable signal asserted (see, col. 6, line 63 to col. 7, line 5), at which point the value of the full instruction address from the program counter register 10 is compared. Accordingly, as discussed above, the alleged detection for detecting a last

> instruction (i.e., the comparison performed by the comparator 14 of Jaggar) is irrelevant to the actual determination of the last instruction, and it is respectfully submitted that the Examiner's position departs from what is disclosed in Jaggar; namely, the actual detection of the last instruction is performed in the branch cache 4 so as to determine a genuine cache hit, and is not performed in the comparator 14 as noted by the fact that branching is not performed even if the lowermost eight bits of the program counter value PC match the value in the reach value latch 16. In this regard, it is important to recognize that if the enable signal of Jaggar is in the ON state, the value of the 111 instruction address from the program counter register 10 is compared in parallel with all of the cache tags so that Jaggar is silent with regard to comparing the %11 instruction address and each of the cache tags one by one.
>
> In contrast, the detecting means of the present invention detects by comparing information on the instruction to be executed with predetermined information on the process one by one."

77. These arguments are not found persuasive for the following reasons:

a) The examiner is not clear how the above quoted statement in the Office Action constitutes an opinion. As is known, a program counter (PC) is inherently incremented each cycle so that a next instruction may be fetched and executed. See column 8, lines 6-11 (note that over multiple clock cycles, the PC holds addresses pointing to instructions A, B, C, etc.). The comparator then compares each PC value (first address &A) with the reach value (address &E). Since there is no match, instruction A is not a last instruction before a branch. The PC, as is known, will then be incremented to hold address &B, which is compared to reach value &E. Like the first time, since there is no match, instruction B is not a last instruction before a branch. Eventually, the PC will be incremented such that it holds address &E. When this is compared to reach value &E, then instruction E will be a last instruction before a branch if a cache hit occurs, i.e., if the PC matches one of the branch tags in the branch cache. It is asserted that this process is not opinion, and it is not understood how this could be viewed as an opinion. In addition, the examiner admits that applicant is correct when saying that a portion of the PC is first compared to a reach value and if a match occurs, then the entire PC is compared to the tags in the cache (i.e., there are multiple comparisons). However, the examiner asserts that this is still equivalent to applicant's

claimed invention. Applicant merely claims a one by one comparison and Jaggar's steps of first comparing a reach value and then comparing branch tags make up said one by one comparison. That is, Jaggar's two comparing steps are still an overall comparison. The overall effect is that the system compares the PC to an instruction address. The initial comparison is used merely to save power by deciding whether or not to enable (power up the cache). In summation, it should be realized though that Jaggar's preliminary comparison and the subsequent comparison happen for each instruction (one by one) and that they are an overall comparison that is divided into two steps.

78. Applicant argues the novelty/rejection of claims 1 and 5 on pages 12-13 of the remarks, in substance that:

> "Moreover, Applicants previously argued, "Jaggar requires that the branch instruction detector 22 detects the last instruction before branching by detecting the branch instruction in order to automatically set the cache data in each cache line 12" and "the present invention does not require such a branch instruction detector for detecting the last instruction before branching any instruction." In response, the Examiner asserted, "Applicant is arguing limitations that are not in the claim" and "it is not clear how the system may initially detect the last instruction before the branch without knowing where the branch is located."
>
> However, similar to the reasons as set forth above, it is not entirely understood which limitations the Examiner alleges as not appearing in the claims. Once again, it appears the Examiner has misinterpreted Applicants' previous arguments. Specifically, in the previous response, Applicants were NOT arguing that the present invention requires a branch instruction detector. Instead, Applicants were arguing that the present invention does not require such a branch instruction detector for detecting the last instruction as demonstrated by Jaggar, and that the Applicants' argument was intended to evidence that Jaggar does not disclose a detecting means that can be reasonably interpreted as the claimed detecting means."

79. These arguments are not found persuasive for the following reasons:

a) When the examiner stated that applicant is arguing limitations which are not in the claims, what was meant is that there is no language within the claim which excludes branch detection. All that is required, according to applicant's claim language, is that a last instruction before a branch is detected. Whether or not this includes detecting a branch instruction in Jaggar is

irrelevant, as applicant's claim uses the term "comprising," which means that extra steps/components may exist in the prior art and still read on the claim (for instance, branch detection).

80. Applicant argues the novelty/rejection of claim 5 on pages 14-15 of the remarks, in substance that:

> "Also, with respect to claim 5, this claim recites in-part wherein the detecting means detects the last instruction based on judgment whether information stored...in the storing means indicates the last instruction. In the pending rejection, the Examiner asserts, "...each instruction address is compared to the predetermined reach value..." However, in formulating this argument, it is clear that the Examiner departs from what is disclosed in Jaggar; namely, the alleged information or "each" instruction address is clearly not stored in the main memory system 8 or the alleged storing means as required by claim 5. Also, the alleged information is merely an instruction address, rather than an information that indicates a last instruction. For these reasons, it is respectfully submitted that claim 5 is patentable over the cited prior art."

81. These arguments are not found persuasive for the following reasons:

a) The unclear claim language does not require that the information be stored in the storing means. The language states "...information indicating a content of **an instruction to be executed** by the instruction executing means **in the storing means**...". The examiner has interpreted the claim such that the instruction to be executed is in the storing means, which is the case in Jaggar. However, the examiner is not conceding that "information" is stored in some storing means. In addition, the instruction address, when in the form of a reach value or a branch tag, is in fact information which indicates a last instruction.

Conclusion

82. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

David J. Huisman
June 14, 2005